4-1988

An Interactive Program for the Analysis of Data from Two-Level Factorial Experiments via Probability Plotting

Stephen V. Crowder
Corning Glass Works

Karen L. Jensen
Iowa State University

W. Robert Stephenson
Iowa State University, wrstephe@iastate.edu

Stephen B. Vardeman
Iowa State University, vardeman@iastate.edu

An Interactive Program for the Analysis of Data from Two-Level Factorial Experiments via Probability Plotting

Abstract

An interactive computer program that expedites the analysis for unreplicated two-level factorial and fractional factorial experimental designs advocated by Daniel (1976) and Box, Hunter, and Hunter (1978) is presented. The program calculates estimated effects via the Yates algorithm, identifies statistically detectable effects via normal plots and half normal plots, fits candidate models via the reverse Yates algorithm, and enables evaluation of candidate models through residual plots. The program can handle the analysis of standard 2p-q fractional factorial experiments where p - q √ 7 and can be modified to allow p - q > 7.

Keywords

Factorial Experiments, Probability Plotting, Yates Algorithm

Disciplines

Statistics and Probability

Comments

This article is published as Crowder, Stephen V., Karen L. Jensen, W. Robert Stephenson, and Stephen B. Vardeman. "An interactive program for the analysis of data from two-level factorial experiments via probability plotting." *Journal of Quality Technology* 20, no. 2 (1988): 140-148. DOI: 10.1080/00224065.1988.11979094.

An Interactive Program for the Analysis of Data from Two-Level Factorial Experiments via Probability Plotting

STEPHEN V. CROWDER

Corning Glass Works, Corning, NY 14831

KAREN L. JENSEN, W. ROBERT STEPHENSON, and STEPHEN B. VARDEMAN

Iowa State University, Ames, IA 50011

An interactive computer program that expedites the analysis for unreplicated two-level factorial and fractional factorial experimental designs advocated by Daniel (1976) and Box, Hunter, and Hunter (1978) is presented. The program calculates estimated effects via the Yates algorithm, identifies statistically detectable effects via normal plots and half normal plots, fits candidate models via the reverse Yates algorithm, and enables evaluation of candidate models through residual plots. The program can handle the analysis of standard 2^{p-q} fractional factorial experiments where $p - q \leqq 7$ and can be modified to allow $p - q > 7$.

Introduction

In Chapters 10 through 13 of Box, Hunter, and Hunter (1978) the authors advocate a powerful method for the analysis of data from two-level factorial and fractional factorial experiments based on the probability plotting device suggested by Daniel (1959, 1976) (see page 74 of Daniel [1976]). The method is especially useful when there is no replication, although much of it can be applied to the analysis of cell sample means from replicated designs.

The method is as follows.

Step 1

A complete two-level factorial design in r factors consists of all 2^r possible combinations of "low" and "high" levels of the individual factors. Similarly, for standard fractional designs involving p factors each at two levels, when 2^{p-q} observations are made there are $r = p - q$ of the p factors such that (ignoring the remaining q factors) all 2^r possible combinations of low and high levels of these factors are represented exactly once in the data set. In either case, one begins by computing the 2^r linear combinations of the observations giving estimated factorial effects (overall mean, main effects, and interactions) of these factors corresponding to a treatment combination where all factors are at their high level. (We will henceforth term a combination where all factors are at their high level an "all-high" combination.) This is most efficiently done using the Yates algorithm.

Step 2

In order to help identify statistically detectable effects from among the 2^r estimates calculated in Step 1, a principle of "effect sparsity" and the device of probability plotting are employed.

It is common in multifactor experiments for a few effects to be much more important than the majority. Indeed, unless this happens, there is little useful information to be gained from recognizing a factorial structure among a group of treatment combinations as opposed to simply treating the various combinations as separate levels of a single omnibus factor. *Effect sparsity* is the term Box, Hunter, and Hunter (1978)

Dr. Crowder is a Senior Statistical Engineer in the Specialty Glass and Ceramics Division. He is a Member of ASQC.

Ms. Jensen is a Graduate Research Assistant in the Statistics Department and the ISU Industry/University Affiliate Program in Productivity, Quality and Reliability.

Dr. Stephenson is an Assistant Professor in the Statistics Department. He is a Member of ASQC.

Dr. Vardeman is a Professor in the Statistics and Industrial Engineering Departments. He is a Member of ASQC.

use for this tendency for a few factorial effects to be dominant.

If one accepts the premise that most factorial effects will typically be small in absolute value compared to a few much larger ones, it follows that in a given experiment most estimated effects will primarily reflect random variation while the few estimated effects corresponding to large or "active" effects will be composed of a larger nonrandom component in addition to random variation. Looking for departures from linearity on a probability plot of estimated effects (caused by a few outlying or relatively extreme values) is a simple graphical way of identifying estimated effects that are not explainable solely in terms of random variation.

Daniel's (1959) original suggestion was to make *half-normal plots* of the *absolute values* of the estimated effects from Step 1. This stemmed from the fact that the use of effects corresponding to the all-high combination in the plotting depends upon an arbitrary designation of levels of the factors as "high" and "low." A different arbitrary designation would change the signs of the plotted estimated effects, and the use of absolute values avoids this problem. Box, Hunter, and Hunter (1978) and Daniel (1976), however, prefer *normal plots* of the *signed* estimated effects from Step 1, arguing that, though they are in a sense arbitrary, the signs carry information that can be useful in understanding the data. (See page 149 of Daniel [1976].)

Step 3

The plotting techniques of Step 2 and subject matter knowledge (plus the aliasing structure for a fractional design) help the investigator identify a candidate set of active effects. It is then useful to generate a fitted mean response for each of the 2^r treatment combinations making use of the estimates from Step 1 for effects judged as active and *assuming all other effects to be zero.* This is efficiently done using the reverse Yates algorithm. When there is more than one candidate set of active effects (perhaps because of "close calls" in the identification process or in a situation where an initial candidate fails subsequent diagnostic checks) fitted mean responses may have to be generated more than once.

Step 4

The generated fitted mean responses can then be compared to the observed responses in order to assess the adequacy of the fitted model. The standard way of making this comparison is to compute *residuals* by subtracting fitted means from corresponding observations. If a fitted model is to be judged adequate, these residuals should appear "patternless," as random variation around zero carrying no useful information. Informative graphical devices for examining residuals include normal plots and plots against variables such as time order of observation, level of experimental factors, and fitted mean response. If the residuals are to be judged as carrying no information, a normal plot should be reasonably linear and plots against other variables should show only random scatter (with reasonably constant variance).

This method of analysis is easy to understand and teach because of its highly graphical nature. However, because several iterations through Steps 3 and 4 may be required, the calculations and plotting can become quite tedious if done by hand. For this reason an interactive computer program was developed.

Program Description

All estimated factorial effects in a complete 2^r design producing observations y are of the form

$$\text{estimated effect} = \frac{1}{2^r} \sum_{l} (\pm y_l) \tag{1}$$

where the sum is over all 2^r values of the index l designating treatment combinations. The signs to be applied to the observations can be determined using easily generated tables discussed, for example, on page 322 of Box, Hunter, and Hunter (1978).

There is some sentiment for the use of 2^{r-1} as the divisor on the right of (1), and in some references including Box, Hunter, and Hunter (1978) the values in display (1) are known as "half effects." Because (1) is consistent with standard definitions for designs where factors have more than two levels we have decided to use the 2^r divisor. However, we stress that the results of this kind of analysis are independent of this choice of a scale factor.

When all 2^r estimated effects corresponding to the all-high combination in a 2^r design are desired, it is most efficient to generate them simultaneously using the Yates algorithm. Nelson (1982) has provided a BASIC program implementing the Yates algorithm. Part of our FORTRAN program is an implementation of the Yates algorithm. Let $\hat{E}$ stand for one of these (signed) estimated effects and $|\hat{E}|$ stand for its absolute value. The program orders the $\hat{E}$'s and $|\hat{E}|$'s and produces probability plots of either all 2^r of these, or all but the one corresponding to the overall mean. (The second choice is more common, but all 2^r values are sometimes plotted, particularly in fractional cases.) When a *normal* plot is requested, the program plots

$$\Phi^{-1}\left(\frac{i-0.5}{n}\right) \quad \text{versus the } i^{\text{th}} \text{ smallest } \hat{E} \text{ being plotted} \tag{2}$$

where n is the number of estimated effects plotted and Φ^{-1} is the inverse of the standard normal distribution function. Since Φ^{-1} cannot be written in closed form, the value of this function for a given cumulative probability value is approximated using Algorithm AS 111, Beasley and Springer (1977). This algorithm appears in the program code as DOUBLE PRECISION FUNCTION PN. When a *half normal plot* is requested, the program instead plots

$$\Phi^{-1}\left(0.5\left(1+\frac{i-0.5}{n}\right)\right) \tag{3}$$

versus the i^{th} smallest $|\hat{E}|$ being plotted.

Pronounced nonlinearity in a plot of the type (2) or (3) is indicative of statistically detectable effects, with points "trailing off" to the left or right on a normal plot or to the right on a half-normal plot corresponding to effects that are detectable.

Once one judges some effects to be active and others not, fitted treatment combination means, $\hat{y}_l$ say, are available by summing for a given treatment combination l, the estimated grand mean and $2^r - 1$ fitted factorial effects corresponding to the particular combination of interest. The program allows these fitted effects to be set to either zero (where the corresponding effect has not been identified as active) or an appropriately chosen plus-or-minus one times the corresponding estimated effect for the all-high combination. That is, fitted means are

$$\hat{y}_l = \sum_j (\pm \hat{F}_j) \tag{4}$$

where the sum is over the 2^r values of j indexing factorial effects, $\hat{F}_j = 0$ where the j^{th} factorial effect has been judged negligible, and $\hat{F}_j = \hat{E}_j$ otherwise. The signs to be applied to the $\hat{F}_j$'s are specific to the combination l being considered. (A sign is + precisely when among the letters naming the main effect or interaction j, there are zero or an even number missing from the letters designating the treatment combination l.)

Simultaneous generation of all 2^r $\hat{y}_l$'s described in (4) is accomplished via reversal of the Yates algorithm. The program then generates 2^r residuals e_l as

$$e_l = y_l - \hat{y}_l$$

and produces both a plot of

$$e_l \text{ versus } \hat{y}_l \tag{5}$$

and a normal plot of the residuals, i.e. a plot of

$$\Phi^{-1}\left(\frac{i-0.5}{2^r}\right) \quad \text{versus the } i^{\text{th}} \text{ smallest } e_l. \tag{6}$$

As discussed in the introduction, these can be used in the process of model checking.

After the plots (5) and (6) are made for a particular choice of fitted effects, the program allows the user to try another choice (without having to reenter the data) or to end the session.

Program Operation

The program first asks the user for the size of the factorial (i.e., for the exponent r in 2^r). It then asks the user for the 2^r observed responses, at each query indicating those factors appearing in the treatment combination at their high level. After computing and displaying the 2^r estimated effects (corresponding to the all-high combination), the program offers the user options to:

1. produce a normal plot of the estimated effects,
2. produce a half-normal plot of the absolute estimated effects,
3. use the reverse Yates algorithm, or
4. exit the program.

If option 1 or 2 is chosen, the user is queried as to the inclusion of the estimated overall mean in the plot, the plot is produced and the user is returned to the main menu. If option 3 is chosen, the user is asked in turn to declare whether each of the fitted effects should be set to zero or should take a value estimated from the "forward" application of the Yates algorithm. After producing the plot of residuals versus fitted combination means and the normal plot of residuals, the program returns the user to the main menu.

As indicated in the Abstract, the present version of the program allows $r \leqq 7$, but could easily be modified to accommodate larger r. The limit on r would then be determined by the amount of data storage allowed by a particular FORTRAN compiler. The necessary changes are

i. array dimensions must be changed from 128 to 2^r in variable declaration statements,
ii. 7 must be changed to r at several places in the main program and in FORMAT statements where A7 occurs, and
iii. characters must be added to the lists in the subroutine STRNGS as follows

```
DATA LTRSC/'a', 'b', 'c', 'd', 'e', 'f', 'g'/
DATA LTRSE/'A', 'B', 'C', 'D', 'E', 'F', 'G'/
```

so that there are r characters in each list.

Example

The data in Table 1 are taken from page 379 of Box, Hunter, and Hunter (1978). They are 16 observations from a (standard) half-fraction of a 2^5 factorial design used to investigate in a chemical reactor study the effects of 5 factors on

$$y = \text{percent reacted.}$$

Notice that by ignoring factor E, each possible combination of levels of factors A, B, C, and D appears exactly once in this experimental design. For analysis purposes these data are entered into the program as if they were observations from an unreplicated full 2^4 factorial experiment in the factors A, B, C, and D. However, the interpretation of estimated effects must be done in light of the *alias structure* of the design, involving *all 5 factors A, B, C, D, and E.* Since this is discussed extensively in Chapters 12 and 13 of Box, Hunter, and Hunter (1978), we will not attempt an explanation here.

The output listing below follows the analysis of these data made in section 12.2 of Box, Hunter, and Hunter (1978). User input appears in boldface type. Data are entered, estimated effects are calculated, a full normal plot of $2^4 - 1 = 15$ estimated effects made, and the reverse Yates algorithm is used setting all fitted effects to zero except those corresponding to the overall mean, the B main effect, the D main effect, the BD interaction, the ABC interaction, and the $ABCD$ interaction. Readers interested in comparing the output to the discussion in Box, Hunter, and Hunter (1978) are reminded that the text uses a 2^{r-1} divisor in (1), so that the present scales will differ from the ones in the text.

TABLE 1. Results of Reactor Study

Letter	Variable	"Low" Level	"High" Level
A	feed rate (l/min)	10	15
B	catalyst (%)	1	2
C	agitation rate (rpm)	100	120
D	temperature (°C)	140	180
E	concentration (%)	3	6

Treatment Combination (Factors at High Level)	Response (% Reacted)
e	56
a	53
b	63
abe	65
c	53
ace	55
bce	67
abc	61
d	69
ade	45
bde	78
abd	93
cde	49
acd	60
bcd	95
abcde	82

Output Listing

```
YATES ALGORITHM

ENTER NUMBER OF FACTORS ( UP TO 7 ):  4

ENTER CELL MEAN FOR SPECIFIED TREATMENT COMBINATION:

(1)     = 56
a       = 53
b       = 63
ab      = 65
c       = 53
ac      = 55
bc      = 67
abc     = 61
d       = 69
ad      = 45
bd      = 78
abd     = 93
cd      = 49
acd     = 60
bcd     = 95
abcd    = 82

ESTIMATED EFFECTS CALCULATED BY YATES ALGORITHM:

I       =   65.2500
A       =   -1.0000
B       =   10.2500
AB      =     .7500
C       =     .0000
AC      =     .2500
BC      =     .7500
ABC     =   -4.7500
D       =    6.1250
AD      =    -.3750
BD      =    5.3750
ABD     =    1.1250
CD      =     .1250
ACD     =     .6250
BCD     =     .6250
ABCD    =   -3.1250

STRIKE CARRIAGE RETURN WHEN YOU WISH TO CONTINUE
```

```
CHOOSE OPTION FOR FURTHER ANALYSIS:

        1.  CONSTRUCT A NORMAL PLOT OF FITTED EFFECTS
        2.  CONSTRUCT A HALF-NORMAL PLOT OF ABSOLUTE FITTED EFFECTS
        3.  USE REVERSE YATES ALGORITHM
        4.  EXIT FROM THE PROGRAM

ENTER NUMBER OF SELECTION:  1


CHOOSE OPTION REGARDING GRAND MEAN:

        1.  INCLUDE GRAND MEAN IN THE PLOT
        2.  DO NOT INCLUDE GRAND MEAN IN THE PLOT

ENTER NUMBER OF SELECTION:  2


                  PLOT OF Y=NORMAL QUANTILE VS. X=FITTED EFFECT

     Y |
       |
  1.83 |                                                                 *
       |
       |                                               *
       |                                            *
   .86 |                          *
       |                        *
       |                       **
       |                       *
  -.12 |                     **
       |                     *
       |                   *
       |                 *
 -1.10 |       *
       |
       |
       | *
 -2.08 |
        -+--------+--------+--------+--------+--------+--------+--------+--------+----
       -4.8     -2.9     -1.0       .9      2.8      4.6      6.5      8.4     10.3  X

STRIKE CARRIAGE RETURN WHEN YOU WISH TO CONTINUE

CHOOSE OPTION FOR FURTHER ANALYSIS:

        1.  CONSTRUCT A NORMAL PLOT OF FITTED EFFECTS
        2.  CONSTRUCT A HALF-NORMAL PLOT OF ABSOLUTE FITTED EFFECTS
        3.  USE REVERSE YATES ALGORITHM
        4.  EXIT FROM THE PROGRAM

ENTER NUMBER OF SELECTION:  3


TO BEGIN THE REVERSE YATES ALGORITHM, SPECIFY WHICH EFFECTS SHOULD BE
USED AS CALCULATED BY THE YATES ALGORITHM, AND WHICH SHOULD BE SET TO ZERO.

I       =   65.2500        SET TO ZERO?  (ENTER Y OR N)  N
A       =   -1.0000        SET TO ZERO?  (ENTER Y OR N)  Y
B       =   10.2500        SET TO ZERO?  (ENTER Y OR N)  N
AB      =     .7500        SET TO ZERO?  (ENTER Y OR N)  Y
C       =     .0000        SET TO ZERO?  (ENTER Y OR N)  Y
AC      =     .2500        SET TO ZERO?  (ENTER Y OR N)  Y
BC      =     .7500        SET TO ZERO?  (ENTER Y OR N)  Y
ABC     =   -4.7500        SET TO ZERO?  (ENTER Y OR N)  N
D       =    6.1250        SET TO ZERO?  (ENTER Y OR N)  N
AD      =    -.3750        SET TO ZERO?  (ENTER Y OR N)  Y
BD      =    5.3750        SET TO ZERO?  (ENTER Y OR N)  N
ABD     =    1.1250        SET TO ZERO?  (ENTER Y OR N)  Y
CD      =     .1250        SET TO ZERO?  (ENTER Y OR N)  Y
ACD     =     .6250        SET TO ZERO?  (ENTER Y OR N)  Y
BCD     =     .6250        SET TO ZERO?  (ENTER Y OR N)  Y
ABCD    =   -3.1250        SET TO ZERO?  (ENTER Y OR N)  N


EFFECT VALUES TO BE USED BY THE REVERSE YATES ALGORITHM:

I       =   65.2500
A       =     .0000
B       =   10.2500
AB      =     .0000
C       =     .0000
AC      =     .0000
BC      =     .0000
ABC     =   -4.7500
D       =    6.1250
AD      =     .0000
BD      =    5.3750
ABD     =     .0000
CD      =     .0000
ACD     =     .0000
BCD     =     .0000
ABCD    =   -3.1250

STRIKE CARRIAGE RETURN WHEN YOU WISH TO CONTINUE
```

TC	CELL MEAN	FITTED CELL MEAN	RESIDUAL
(1)	56.0000	55.8750	.1250
a	53.0000	52.6250	.3750
b	63.0000	62.3750	.6250
ab	65.0000	65.6250	-.6250
c	53.0000	52.6250	.3750
ac	55.0000	55.8750	-.8750
bc	67.0000	65.6250	1.3750
abc	61.0000	62.3750	-1.3750
d	69.0000	63.6250	5.3750
ad	45.0000	47.8750	-2.8750
bd	78.0000	79.1250	-1.1250
abd	93.0000	94.8750	-1.8750
cd	49.0000	47.8750	1.1250
acd	60.0000	63.6250	-3.6250
bcd	95.0000	94.8750	.1250
abcd	82.0000	79.1250	2.8750

```
STRIKE CARRIAGE RETURN WHEN YOU WISH TO CONTINUE


                   PLOT OF Y-RESIDUAL VS. X-FITTED CELL MEAN

     Y |
       |
  5.38 |                      *
       |
       |
       |
  2.98 |                                            *
       |
       |                         *
       | *                  *
   .58 |       *   *                                                      *
       |
       |           *             *                  *
       |                    *
 -1.83 |                                                                  *
       | *
       |
       |                      *
 -4.23 |
        -+-------+--------+--------+--------+--------+--------+--------+--------+----
       47.9    53.8     59.6     65.5     71.4     77.3     83.1     89.0     94.9  X

STRIKE CARRIAGE RETURN WHEN YOU WISH TO CONTINUE


                   PLOT OF Y=NORMAL QUANTILE VS. X=RESIDUAL

     Y |
       |
  1.86 |                                                                  *
       |
       |                                                *
       |                                     *
   .87 |                                   *
       |                              * *
       |                              *
       |                            *
  -.12 |                      *
       |                   * *
       |                 *
       |             *
 -1.12 |      *
       |
       | *
       |
 -2.11 |
        -+--------+--------+--------+--------+--------+--------+--------+--------+----
       -3.6     -2.5     -1.4      -.3       .9      2.0      3.1      4.3      5.4  X

STRIKE CARRIAGE RETURN WHEN YOU WISH TO CONTINUE


CHOOSE OPTION FOR FURTHER ANALYSIS:

        1.  CONSTRUCT A NORMAL PLOT OF FITTED EFFECTS
        2.  CONSTRUCT A HALF-NORMAL PLOT OF ABSOLUTE FITTED EFFECTS
        3.  USE REVERSE YATES ALGORITHM
        4.  EXIT FROM THE PROGRAM

ENTER NUMBER OF SELECTION:  4


Stop - Program terminated.
```

References

BEASLEY, J. D. and SPRINGER, S. G. (1977). "The Percentage Points of the Normal Distribution." *Applied Statistics* 26, pp. 118–121.

BOX, G. E. P.; HUNTER, W. G.; and HUNTER, J. S. (1978). *Statistics for Experimenters.* John Wiley & Sons, New York, NY.

DANIEL, C. (1959). "Use of Half-Normal Plots in Interpreting Factorial Two-Level Experiments." *Technometrics* 1, pp. 311–341.

DANIEL, C. (1976). *Applications of Statistics to Industrial Experimentation.* John Wiley & Sons, New York, NY.

NELSON, L. S. (1982). "Analysis of Two-Level Factorial Experiments." *Journal of Quality Technology* 14, pp. 95–98.

Key Words: *Factorial Experiments, Probability Plotting, Yates Algorithm.*

Program Listing

```
      REAL*8 C(128),EFF(128),PN
      INTEGER R,N,CH1
      CHARACTER*7 TC(128),TE(128)
C
      WRITE(*,100)
  100 FORMAT(////,25X,'YATES ALGORITHM',////)
C
  102 WRITE(*,104)
  104 FORMAT(1X,'ENTER NUMBER OF FACTORS ',
     +'( UP TO 7 ):  ',\)
      READ(*,*) R
C
C  Check for error in entering number of factors
C
      IF (R.GE.1.AND.R.LE.7) GOTO 110
      WRITE(*,105)
  105 FORMAT(///,1X,'NUMBER OF FACTORS MUST BE ',
     +'BETWEEN 1 AND 7',/,
     +1X,'PLEASE TRY AGAIN')
      GOTO 102
C
C  Create character strings representing treatment
C  combinations and treatment effects.
C
  110 CALL STRNGS(R,TC,TE)
C
C  Read cell means from keyboard.
C
      N=2**R
      WRITE(*,115)
  115 FORMAT(///,1X,'ENTER CELL MEAN FOR SPECIFIED',
     +' TREATMENT COMBINATION:',/)
      DO 125 I=1,N
        WRITE(*,120) TC(I)
  120   FORMAT(1X,A7,' = ',\)
        READ(*,*) C(I)
        EFF(I)=C(I)
  125 CONTINUE
C
C  Perform Yates algorithm and display results
C
      CALL YATES(R,EFF)
      WRITE(*,130)
  130 FORMAT(///,1X,'ESTIMATED EFFECTS CALCULATED',
     +' BY YATES ALGORITHM:',/)
      DO 140 I=1,N
        WRITE(*,135) TE(I),EFF(I)
  135   FORMAT(1X,A7,1X,'=',1X,F9.4)
  140 CONTINUE
      CALL PAUSE
C
C  Perform further analyses
C
  300 WRITE (*,301)
  301 FORMAT(///,1X,'CHOOSE OPTION FOR FURTHER ',
     +'ANALYSIS:',
     +//,10X,'1.  CONSTRUCT A NORMAL PLOT OF FITTED ',
     +'EFFECTS',
     +/,10X,'2.  CONSTRUCT A HALF-NORMAL PLOT OF ',
     +'ABSOLUTE FITTED EFFECTS',
     +/,10X,'3.  USE REVERSE YATES ALGORITHM',
     +/,10X,'4.  EXIT FROM THE PROGRAM',
     +//,1X,'ENTER NUMBER OF SELECTION:  ',\)
      READ(*,*) CH1
      IF(CH1.LT.1.OR.CH1.GT.4) GOTO 340
C
      GOTO (310,310,330,350),CH1
C
  310 CALL PLOTS(CH1,N,EFF)
      GOTO 300
C
  330 CALL RYATES(N,R,EFF,TC,TE,C)
      GOTO 300
C
  340 WRITE(*,345)
  345 FORMAT(///,1X,'INVALID SELECTION, PLEASE TRY ',
     +'AGAIN')
      GOTO 300
C
  350 WRITE(*,355)
  355 FORMAT(////)
      STOP
      END
C
C
C-----------------------------------------------------
C
      SUBROUTINE STRNGS(R,TC,TE)
C
      INTEGER R,NUM,BIN(7),DECIML
      CHARACTER*7 TC(128),TE(128),STRNGC,STRNGE
      CHARACTER*1 ARRAYC(7),ARRAYE(7),LTRSC(7),
     +LTRSE(7)
C
      EQUIVALENCE (STRNGC,ARRAYC),(STRNGE,ARRAYE)
C
      DATA LTRSC/'a','b','c','d','e','f','g'/
      DATA LTRSE/'A','B','C','D','E','F','G'/
C
C  Create the character strings that represent
C  treatment combinations and treatment effects by
C  using the binary representation of a number to
C  determine which letters appear in the string.
C
      NUM=2**R
      TC(1)='(1)'
      TE(1)='I'
      DO 200 I=1,NUM-1
C
C  First, convert a decimal number into binary
C
        DECIML=I
        DO 210 J=1,R
          N=2**(R-J)
          L=R-J+1
          IF (DECIML.GE.N) THEN
            BIN(L)=1
            DECIML=DECIML-N
          ELSE
            BIN(L)=0
          ENDIF
  210   CONTINUE
C
C  Now build the character strings associated with
C  the binary number.  First blank out the character
C  string,
C
        DO 220 J=1,7
          ARRAYC(J)=' '
          ARRAYE(J)=' '
  220   CONTINUE
C
C  then fill in the appropriate letters.
C
        K=1
        DO 230 J=1,R
          IF (BIN(J).EQ.0) GOTO 230
          ARRAYC(K)=LTRSC(J)
          ARRAYE(K)=LTRSE(J)
          K=K+1
  230   CONTINUE
C
C  Finally, put the strings into the arrays to be
C  used in the program.
C
        TC(I+1)=STRNGC
        TE(I+1)=STRNGE
  200 CONTINUE
C
```

```
      RETURN
      END
C
C
C-------------------------------------------------------
C
      SUBROUTINE YATES(R,EFF)
C
      REAL*8 EFF(128),STO(128)
      INTEGER I1,R,N
C
      N=2**R
      I1=2**(R-1)
C
      DO 83Ø I=1,R
        DO 81Ø J=1,I1
          STO(J)=EFF(2*J-1) + EFF(2*J)
          STO(J+I1)=EFF(2*J) - EFF(2*J-1)
  81Ø   CONTINUE
        DO 82Ø J=1,N
          EFF(J)=STO(J)
  82Ø   CONTINUE
  83Ø CONTINUE
C
      DO 84Ø I=1,N
        EFF(I)=EFF(I)/N
  84Ø CONTINUE
C
      RETURN
      END
C
C-------------------------------------------------------
C
      SUBROUTINE PLOTS(TYPE,NUM,EFF)
C
      INTEGER TYPE,NUM,CH2,NN
      REAL*8 EFF(128),P(128)
C
  4Ø4 WRITE(*,4Ø5)
  4Ø5 FORMAT(///,1X,'CHOOSE OPTION REGARDING GRAND ',
     +'MEAN:',
     +//,1ØX,'1.  INCLUDE GRAND MEAN IN THE PLOT',
     +/,1ØX,'2.  DO NOT INCLUDE GRAND MEAN IN THE ',
     +'PLOT',
     +//,1X,'ENTER NUMBER OF SELECTION:  ',\)
      READ(*,*) CH2
      IF (CH2.LT.1.OR.CH2.GT.2) GOTO 43Ø
C
      GOTO (41Ø,42Ø),CH2
C
  41Ø CALL NPLOT(TYPE,NUM,EFF)
      GOTO 49Ø
C
  42Ø DO 425 I=2,NUM
        P(I-1)=EFF(I)
  425 CONTINUE
      NN=NUM-1
      CALL NPLOT(TYPE,NN,P)
      GOTO 49Ø
C
  43Ø WRITE(*,435)
  435 FORMAT(///,1X,'INVALID SELECTION, PLEASE TRY
     +'AGAIN')
      GOTO 4Ø4
C
  49Ø RETURN
      END
C
C-------------------------------------------------------
C
      SUBROUTINE NPLOT(TYPE,NUM,VALS)
C
      INTEGER TYPE,NUM
      REAL*8 VALS(128),AVAL(128),SORTV(128),
     +SORTA(128),ARG,PN,Y(128)
C
      GOTO (51Ø,53Ø),TYPE
C
C  Contruct a normal plot
C
  51Ø DO 52Ø I=1,NUM
         ARG=(DBLE(I)-.5ØDØ)/DBLE(NUM)
         Y(I)=PN(ARG)
  52Ø CONTINUE
      CALL SORT(NUM,VALS,SORTV)
      WRITE(*,525)
  525 FORMAT(///,2ØX,'PLOT OF Y=NORMAL QUANTILE ',
     +'VS. X=FITTED EFFECT')
      CALL PLTTR(NUM,SORTV,Y)
      GOTO 56Ø
C
C  Construct a half-normal plot
C
  53Ø DO 54Ø I=1,NUM
         ARG=(DBLE(I)-.5ØDØ)/DBLE(NUM)
         Y(I)=PN((ARG/2.ØDØ) + .5ØDØ)
         AVAL(I)=DABS(VALS(I))
  54Ø CONTINUE
      CALL SORT(NUM,AVAL,SORTA)
      WRITE(*,545)
  545 FORMAT(///,13X,'PLOT OF Y=HALF-NORMAL ',
     +'QUANTILE VS. X=ABSOLUTE FITTED EFFECT')
      CALL PLTTR(NUM,SORTA,Y)
C
  56Ø RETURN
      END
C
C-------------------------------------------------------
C
      SUBROUTINE RYATES(N,R,EFF,TC,TE,C)
C
      INTEGER N,R
      REAL*8 EFF(128),EFFR(128),Y(128),T(128),
     +RES(128),C(128),PN,ARG,SORTR(128)
      CHARACTER*7 TC(128),TE(128)
C
C  Use subroutine INPUT to specify values for effects,
C  either the estimated value from Yates algorithm or
C  zero.  Estimated values from Yates algorithm are in
C  EFF, revised values to be used in reverse Yates
C  algorithm are put in EFFR.
C
      CALL INPUT(N,EFF,TE,EFFR)
C
C  Calculate fitted cell means by reverse Yates
C  algorithm, calculate residuals, and display results
C
      CALL YATES(R,EFFR)
      DO 61Ø I=1,N
        T(I)=EFFR(N-I+1)*(2**R)
  61Ø CONTINUE
      WRITE(*,615)
  615 FORMAT(///,12X,'TC',9X,'CELL MEAN',5X,
     +'FITTED CELL MEAN',5X,'RESIDUAL',/)
      DO 625 I=1,N
        EFFR(I)=T(I)
        RES(I)=C(I)-EFFR(I)
        WRITE(*,62Ø) TC(I),C(I),EFFR(I),RES(I)
  62Ø   FORMAT(11X,A7,5X,F9.4,8X,F9.4,9X,F9.4)
  625 CONTINUE
C
      CALL PAUSE
C
C  Display residual plot and normal plot
C
      WRITE(*,63Ø)
  63Ø FORMAT(///,2ØX,'PLOT OF Y=RESIDUAL VS. ',
     +'X=FITTED CELL MEAN')
      CALL PLTTR(N,EFFR,RES)
C
      CALL SORT(N,RES,SORTR)
      DO 635 I=1,N
```

```
        ARG=(DBLE(I)-.50D0)/DBLE(N)
        Y(I)=PN(ARG)
  635 CONTINUE
      WRITE(*,640)
  640 FORMAT(///,20X,'PLOT OF Y=NORMAL QUANTILE ',
     +'VS. X=RESIDUAL')
      CALL PLTTR(N,SORTR,Y)
C
  645 RETURN
      END
C
C
C------------------------------------------------------
C
      SUBROUTINE INPUT(N,ORIG,TE,REV)
C
      INTEGER N
      CHARACTER*7 TE(128)
      CHARACTER*1 CH
      REAL*8 ORIG(128),REV(128)
C
      WRITE(*,700)
  700 FORMAT(///,1X,'TO BEGIN THE REVERSE YATES ',
     +'ALGORITHM, SPECIFY WHICH EFFECTS SHOULD BE',
     +/,1X,'USED AS CALCULATED BY THE YATES ',
     +'ALGORITHM, AND WHICH SHOULD BE SET TO ZERO.',/)
C
      DO 730 I=1,N
  705   WRITE(*,710) TE(I),ORIG(I)
  710   FORMAT(1X,A7,' = ',F9.4,10X,
     +'SET TO ZERO?  (ENTER Y OR N)  ',\)
        READ(*,715) CH
  715   FORMAT(A1)
        IF (CH.NE.'N'.AND.CH.NE.'n'.AND.
     +CH.NE.'Y'.AND.CH.NE.'y') GOTO 720
        IF (CH.EQ.'N'.OR.CH.EQ.'n') REV(N-I+1)=ORIG(I)
        IF (CH.EQ.'Y'.OR.CH.EQ.'y') REV(N-I+1)=0.0D0
        GOTO 730
C
C  Note that the order of the effect values is
C  reversed
C
  720   WRITE(*,725)
  725   FORMAT(/,1X,'INVALID SELECTION, PLEASE TRY ',
     +'AGAIN')
        GOTO 705
C
  730 CONTINUE
C
      WRITE(*,735)
  735 FORMAT(///,1X,'EFFECT VALUES TO BE USED',
     +' BY THE REVERSE YATES ALGORITHM:',/)
      DO 750 I=1,N
        WRITE(*,740) TE(I),REV(N-I+1)
  740   FORMAT(1X,A7,1X,'=',1X,F9.4)
  750 CONTINUE
C
      CALL PAUSE
C
      RETURN
      END
C
C------------------------------------------------------
C
      SUBROUTINE SORT(N,OBS,SOBS)
C
C  Sort algorithm due to Loeser, Communications of
C  the ACM, Vol. 17, No. 3, page 143.
C
      REAL*8 OBS(N),SOBS(N),S
C
      DO 190 J=1,N
        SOBS(J)=OBS(J)
  190 CONTINUE
C
      I=1
  101 IF (I-N) 102,102,103
  102 I=I+I
      GO TO 101
  103 M=I-1
  104 M=M/2
      IF (M) 110,110,105
  105 K=N-M
      DO 109 J=1,K
        I=J+M
  106   I=I-M
        IF (I) 109,109,107
  107   L=I+M
        IF (SOBS(L)-SOBS(I)) 108,108,109
  108   S=SOBS(I)
        SOBS(I)=SOBS(L)
        SOBS(L)=S
        GO TO 106
  109 CONTINUE
      GO TO 104
  110 RETURN
      END
C
C------------------------------------------------------
C
      DOUBLE PRECISION FUNCTION PN(P)
C
C  Algorithm AS 111, Applied Statistics (1977),
C  vol. 26, pp 118-121.
C
      REAL*8 A0,A1,A2,A3,B1,B2,B3,B4,C0,C1,C2,C3,D1,
     +D2,P,Q,R
      A0=2.50662823884D0
      A1=-18.61500062529D0
      A2=41.39119773534D0
      A3=-25.44106049637D0
      B1=-8.47351093090D0
      B2=23.08336743743D0
      B3=-21.06224101826D0
      B4=3.13082909833D0
      C0=-2.78718931138D0
      C1=-2.29796479138D0
      C2=4.85014127135D0
      C3=2.32121276858D0
      D1=3.54388924762D0
      D2=1.63706781897D0
      Q=P-.50D0
      IF (DABS(Q) .GT. .42D0) GO TO 100
      R=Q*Q
      PN=Q*(((A3*R + A2)*R + A1)*R + A0)/
     +((((B4*R + B3)*R + B2)*R + B1)*R + 1.0D0)
      RETURN
C
  100 R=P
      IF (Q .GT. 0.0D0) R=1.0D0-P
      R=DSQRT(-DLOG(R))
      PN=(((C3*R + C2)*R +C1)*R + C0)/
     +((D2*R + D1)*R + 1.0D0)
      IF (Q .LT. 0.0D0) PN=-PN
      RETURN
C
      END
C
C
C------------------------------------------------------
C
      SUBROUTINE PLTTR(N,HORIZ,VERT)
C
      INTEGER I,J,K,COUNT,PRNTFLG(75)
      REAL*8 VERT(128),HORIZ(128),UP,LO,MX,MN,X(128),
     +DX,DY,XMIN,XMAX
      CHARACTER*1 PRNTLN(68),PRNTCHR(6),GO
      DATA PRNTCHR/' ','*','*','*','*','*'/
C
C  Find largest and smallest horizontal coordinates
C  and determine the horizontal scaling factor, DX.
C
```

```
      MN=9999999999.9D0
      MX=-9999999999.9D0
      DO 900 I=1,N
       IF (HORIZ(I) .GT. MX) MX=HORIZ(I)
       IF (HORIZ(I) .LT. MN) MN=HORIZ(I)
  900 CONTINUE
      IF (MX.NE.MN) GOTO 902
      MX=MX+1.0D0
      MN=MN-1.0D0
  902 XMIN=MN
      XMAX=MX
      DX=(MX-MN)/64.0D0
C
C  Do the same for the vertical scale.
C
      MN=9999999999.9D0
      MX=-9999999999.9D0
      DO 905 I=1,N
       IF (VERT(I) .GT. MX) MX=VERT(I)
       IF (VERT(I) .LT. MN) MN=VERT(I)
  905 CONTINUE
      IF (MX.NE.MN) GOTO 908
      MX=MX+1.0D0
      MN=MN-1.0D0
  908 DY=(MX-MN)/15.0D0
C
C  Draw the plot a line at a time.
C
      WRITE (*,910)
  910 FORMAT(/,'   Y',5X,'|',/,9X,'|')
      LO=MX
      COUNT=3
C
      DO 940 I=1,17
        DO 915 J=1,65
         PRNTFLG(J)=1
  915   CONTINUE
        UP=LO
        LO=UP-DY
        DO 917 K=1,N
          IF ((VERT(K) .GT. LO) .AND.
     +        (VERT(K) .LE. UP)) THEN
            J=INT(SNGL((HORIZ(K)-XMIN)/DX))+1
            PRNTFLG(J)=PRNTFLG(J)+1
            IF (PRNTFLG(J) .GT. 6) PRNTFLG(J)=6
          END IF
  917   CONTINUE
        IF (COUNT.EQ.3) THEN
          WRITE (*,920) UP,
     +                  (PRNTCHR(PRNTFLG(J)),J=1,65)
  920     FORMAT (1X,F8.2,'| ',65A1)
          COUNT=0
        ELSE
          WRITE (*,925) (PRNTCHR(PRNTFLG(J)),J=1,65)
  925     FORMAT (9X,'| ',65A1)
          COUNT=COUNT+1
        ENDIF
  940 CONTINUE
C
C  Put the labels under the horizontal axis.
C
      WRITE (*,945)
  945 FORMAT (10X,'-+-------+-------+-------+----',
     +'---+-------+-------+-------+----')
      DX=(XMAX-XMIN)/8.0D0
      DO 950 I=1,9
        X(I)=XMIN+DBLE(I-1)*DX
  950 CONTINUE
C
      WRITE (*,960) (X(I),I=1,9)
  960 FORMAT(7X,9(F6.1,2X),'X')
C
      CALL PAUSE
C
      RETURN
      END
C
C---------------------------------------------------
C
      SUBROUTINE PAUSE
C
      CHARACTER*1 GO
C
C  Pause to let the user view a display.
C
      WRITE(*,70)
   70 FORMAT(/,1X,'STRIKE CARRIAGE RETURN ',
     +'WHEN YOU WISH TO CONTINUE  ',\)
      READ(*,75) GO
   75 FORMAT(A1)
C
      RETURN
      END
C
C---------------------------------------------------
```